# INTERAK COMPUTER SYSTEM

Newsletter for Interak & PC Users.

Edited by Patrick Meehan.

Interak Computer Supplied by :-
Greenbank Electronics.

# Interak Newsletter.

This is the first of what hopefully will be a regular newsletter for Interak and IBM PC users. Since the demise of the old newsletter several developments have occurred concerning the development of Interak and even Greenbank Electronics itself.

David Parkins of Greenbank has recently decided to sell IBM PC's as well as the Interak system. The PC compatibles he stocks are sold under the badge of 'Grel Systems'. Ask David or John Parkins of Greenbank Electronics for a price list.

On the Interak front both David and myself have working CPU boards using Zilog's latest Z80 compatible processor the Z80280, usually known as the Z280. This new CPU has the potential to greatly increase Interak's capabilities. Below is listed a few of the goodies to be found within the Z280:-

1. 16MB address range controlled by an onboard memory management unit.

2. Onboard 256 bytes of full associative cache ram.

3. 4 DMA channels; these can be controlled via software only as well as by the normal hardware handshaking.

4. 3 counter timers.

5. Vastly increased instruction set.

When the cache ram is enabled the Z280, running the system bus clock at 4MHz, runs approximately at twice the speed of a Z80A.

As can be seen this gives the Interak considerable power for all sorts of experimentation not possible before ie multi-tasking industrial control. Hopefully enough interest will be generated so that a PCB will be produced for sale to users. At the moment both David's and my CPU board are in prototype form. There has been rumours

that Zilog is in the process of designing a Z80380. With the 64180 and Z80180 there seems to be plenty of activity in Z80 baesd designs.

Also on the Interak front is a graphics card using the Texas Instrument's chip the 34061. The prototype version I have has a resolution of 512 x 256 x 16 colours in mode 1 and 478 x 256 x 4096 colours in mode 2. This card known as the CGV-12 does have a PCB layout created and has also been improved so that IBM PC VGA monitors can be used instead of the old PAL based colour monitors (CGA), thus the resolution has been increased to 640 x 480. Once again I hope that enough interest can be generated so that David will produce a PCB for the CGV-12.

For those of you who are interested in industrial control systems, Greenbank can supply a Z80 SBC (Single Board Computer). The SBC comprises of external ROM/RAM, to a maximum of 32KB each, the ram can also be battery backed, also a Dallas Semiconductor's RTC (Real Time Clock) can be fitted. Intel's ubiquitous 8255 is used for external interfacing; two of these being fitted thus giving 48 digital control lines. The price is extremely competitive at around £79.00. Development can be carried out on Interak before blowing your final code into EPROM. Below is a simple control program in 'C' that was used to control the speed of a stepper motor. The compiler used was Hi-Soft's 'C' compiler and can be supplied by Greenbank Electronics.

```
#data          0x3000
#include       <stdio.h>
#define        PORTA   0x24
#define        PORTB   0x25
#define        PORTC   0x26
#define        CNTRL   0x27
#define        MODE0   0x80
#define        HI_PULSE        0x1E
#define        LO_PULSE        0x1A

/*****************************/
/* program to rotate stepper  */
/* motor using SAA 1027 driver*/
```

```
/* IC via 8255 on SBC-1. motor*/
/* and driver circuit as used */
/* on IBM clones i/o card.    */
/*****************************/

main()
{
int bk_lash;

 set_reg();    /*setup 8255 PIA*/
 fst_pulse();  /*rotate motor quickly*/
 /* below loop to allow mechanical backlash to settle*/
 for(bk_lash=0; bk_lash<1000; bk_lash++);
 slo_pulse();  /*rotate motor slowly*/
} /*closing braces of main*/

set_reg()
{
 out(MODE0, CNTRL);
} /*closing braces of set_reg*/

fst_pulse()
{
 int loop;

  for(loop=0; loop<1000; loop++)
       {
        out(HI_PULSE, PORTB);
        delay_1();    /*delay to  set motor speed*/
        out(LO_PULSE, PORTB);
        delay_1();
       } /*closing braces of for()*/
} /*closing braces of fst_pulse()*/
```

```
slo_pulse()
{
 int loop;

  for(loop=0; loop<450; loop++)
       {
        out(HI_PULSE, PORTB);
        delay_2();    /*delay to  set motor speed*/
        out(LO_PULSE, PORTB);
        delay_2();
       } /*closing braces of for()*/
} /*closing braces of slo_pulse*/

delay_1()
{
 int count;

  for(count=0; count<70; count++);
} /*closing braces of delay()*/

delay_2()
{
 int count;

  for(count=0; count<300; count++);
} /*closing braces of delay()*/

#include ?stdio.lib?
```

C++.

I have recently started to investigate Borland's Turbo C++ programming language for the IBM PC. The user interface is extremely friendly and at the same time powerful. Multiple windows can be opened up to allow editing of more than one source file. Those

of you who are 'C' programmers will be pleased at some of the improvements that have been added to 'C++', eg :-

```
for(int i=0; i<9; i++)
```

Notice that the variable 'i' is declared within the for loop. 'C++' allows you to define variables anywhere in a block of code rather than having to declare them at the start of a function declaration. In the example above 'i' will only be seen by the code within the for loop.

I am presently taking a course on 'C++' and object oriented programming so I am unable to go into classes and objects. Once I have gained more experience I will write on this subject further.

That about wraps it up for this short but hopefully interesting newsletter.

14/11/92.